Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 20, 2011

J. Bi
CERNET
G. Yao
Tsinghua University
J. Halpern
Newbridge Networks Inc
E. Levy-Abegnoli
Cisco Systems

November 16, 2010

SAVI for Mixed Address Assignment Methods Scenario <draft-bi-savi-mix-03.txt>

Abstract

This document reviews how multiple address discovery methods can coexist in a single savi device and collisions are resolved when the same binding entry is discovered by two or more methods.

Table of Contents

1. Introduction

There are currently several documents [I-D.ietf-savi-fcfs], [I-D.ietf-savi-dhcp], [I-D.ietf-savi-send] that describe the different methods by which a switch can discover and record bindings between a node's layer3 address and a binding anchor and use that binding to perform Source Address Validation.

The method used by nodes to assign the address drove the break down into these multiple documents, whether StateLess Autoconfiguration (SLACC), Dynamic Host Control Protocol (DHCP), Secure Neighbor Discovery (SeND) or manual. Each of these documents describes separately how one particular discovery method deals with address collisions.

While multiple assignment methods can be used in the same layer2 domain, a savi-switch might have to deal with a mix of binding discovery methods. The purpose of this document is to provide recommendations to avoid collisions and to review collisions handling when two or more such methods come up with competing bindings.

2. Terminology

3. Mixed Address Assignment Methods Scenario

Currently, there are four SAVI solutions which cover different types of address assignment methods:

1. SAVI-FCFS: SLAAC
2. SAVI-DHCP: stateful DHCP, static DHCP
3. SAVI-SeND: CGA with certificate, CGA without certificate
4. Manually configuration: static address manually configured by administrator on SAVI device.

Any combination of address assignment methods can be potentially found within a layer2 domain, and a savi device will have to implement the corresponding savi discovery methods (savi solutions) to prevent packets from valid sources to be filtered out. If more than one SAVI solution is enabled on a SAVI device, the method is referred to as "mix address assignment method" in this document.

4. Basic Structure

Different savi solutions are independent from each other, each one handling its own entries. In the absence of a reconciliation, each solution will reject packets sourced with an address it did not

discovered. To prevent addresses discovered by one solution to be filtered out by another, the binding table should be shared by all the solutions. However this could create some conflict when the same entry is discovered by two different methods: the main purpose of this document is to resolve such conflicts if and when they happen.

5. Problem Scope, Statement and Solution

5.1. Problem Scope

This document reviews the case of collisions between different SAVI solutions. Collision happening within a given solution is not in the scope of this document.

5.2. Recommendations for preventing collisions

If each solution has a dedicated address space, collisions won't happen. Thus, it is recommended to avoid overlap in the address space across SAVI solutions enabled on any particular savi switch. More specifically:

1. DHCP/Static: exclude the static address from the DHCP pool.
2. DHCP/SLAAC: separate the prefix scope of DHCP and SLAAC. Set the A bit in Prefix information option of Router Advertisement for SLAAC prefix. And set the M bit in Router Advertisement for DHCP prefix. [RFC4861] [RFC4862].
3. SLAAC/Static: separate the prefix scope of SLAAC and Static. It may be impossible in practice. SAVI device can perform DAD proxy for static address to hold the address from SLAAC node.
4. SEND/non-SEND: In an environment where SeND is deployed, the only way to avoid collisions in the SAVI devices is to have SeND-only nodes. In a mixed environment, two nodes, SeND and non-SeND, could configure the same address and the SAVI-device will have to deal with a collision.

5.3. Binding on the Same Address

In situation where collisions could not be avoided, two cases should be considered:

1. The same address is bound on two different binding anchors by different SAVI solutions.
2. The same address is bound on the same binding anchor by different SAVI solutions.

5.3.1. Same Address on Different Binding Anchors

5.3.1.1. Basic preference

Within the SAVI perimeter, one address bound to a binding anchor by one SAVI solution could also be bound by another SAVI solution to a different binding anchor. For example an address could be initially bound to a binding anchor by SAVI-FCFS solution. If another host is assigned the same address from DHCP and the DAD procedure is not performed, the same address will also be bound to the new binding anchor. Both bindings are legitimate in the corresponding solution.

Though it is possible that the hosts and network can still work in such scenario, the uniqueness of address is not assured. The SAVI device must decide whom the address should be bound with. A binding preference level based solution is proposed here.

To determine a proper preference level, following evidences are used:

1. "Duplicate Address Detection MUST be performed on all unicast addresses prior to assigning them to an interface, regardless of whether they are obtained through stateless autoconfiguration, DHCPv6, or manual configuration,..." [RFC4862]
2. "A tentative address that is determined to be a duplicate as described above MUST NOT be assigned to an interface,..." [RFC4862]
3. "The client SHOULD perform duplicate address detection on each of the addresses in any IAs it receives in the Reply message before using that address for traffic." [RFC3315]
4. "A SEND node that uses the CGA authorization method to protect Neighbor Solicitations SHOULD perform Duplicate Address Detection as follows. If Duplicate Address Detection indicates that the tentative address is already in use, the node generates a new tentative CGA. If after three consecutive attempts no non-unique address is generated, it logs a system error and gives up attempting to generate an address for that interface.

   When performing Duplicate Address Detection for the first tentative address, the node accepts both secured and unsecured Neighbor Advertisements and Solicitations received in response to the Neighbor Solicitations. When performing Duplicate Address Detection for the second or third tentative address, it ignores unsecured Neighbor Advertisements and Solicitations." [RFC3971]
5. "The node MAY have a configuration option whereby it ignores unsecured advertisements, even when performing Duplicate Address Detection for the first tentative address. This configuration option SHOULD be disabled by default. This is a recovery mechanism for cases in which attacks against the first address become common." [RFC3971]

From the above materials, FCFS is found to be a universal principle with only one exception: SEND node may use a duplicate address if the DAD NA is only from non-SEND node. And Duplicate Address Detection is enforced to detect the uniqueness of address (though in [RFC3315], "SHOULD" is used but not "MUST"). The static address is not covered in any document, as we believe the "manual configuration" in [RFC4862] means address configured on host by user, but not static address must be protected for servers and special purpose.

The following preference level can be inferred from listed materials and above analysis:

1. SLAAC, DHCP and manually configured address by user have the same priority.
2. SEND can have higher priority because it may configure an address bound by non-SEND node.
3. Static address should have the highest priority to ensure administrator having the right to manage the usage of address.

Combined solution preference with binding sequence, there will be 16 scenarios (Denote solutions by FCFS, DHCP, SEND, and Admin correspondingly):

Existing	Candidate	PREFERENCE
FCFS	FCFS	In the scope of SAVI-SLAAC
FCFS	DHCP	FCFS
FCFS	SEND	SEND
FCFS	Admin	Admin
DHCP	FCFS	DHCP
DHCP	DHCP	In the scope of SAVI-DHCP
DHCP	SEND	SEND
DHCP	Admin	Admin
SEND	FCFS	SEND
SEND	DHCP	SEND
SEND	SEND	In the scope of SAVI-SEND
SEND	Admin	Admin
Admin	FCFS	Admin
Admin	DHCP	Admin
Admin	SEND	Admin
Admin	Admin	Candidate binding

5.3.1.2. Issues in Multiple SAVI Device Scenario

A single SAVI device doesn't have the information of all bound addresses on the perimeter. Therefore a collision may not be explicit based only on local bindings. To make the perimeter-scope collision explicit to each SAVI device requires:

1. A SAVI device must have the ability to know whether a local binding setup request violate a binding on other SAVI devices or not.
2. A SAVI device must have the ability to know whether a local binding should be removed because the address is bound on another SAVI device by solution with higher priority.

The first requirement is relatively easy to meet, as DAD must have been performed on address bound by SAVI-SLAAC and SAVI-SEND, and there is no need to check if a static address violates an existing binding. However DAD is not required by SAVI-DHCP, and static addresses must be prevented from being grabbed by other solutions. Thus, following mechanisms MUST be enforced:

1. SAVI device MUST perform DAD procedure on DHCP address or track if DAD performed by DHCP client itself is successful before binding a DHCP address. Only if the DAD succeeds, the DHCP address can be bound.
2. SAVI device MUST perform DAD proxy for static address. Or all the other SAVI devices MUST be configured to deny static address bound on other SAVI devices, in condition that SAVI-SEND is enabled and it may bind a static address.
3. The second requirement is relatively hard to satisfy. Whenever SAVI-SEND decides to bind an address even it is used by a non-SEND node, and a bound address is bound manually to another binding anchor, the SAVI device with the existing binding must get noticed and delete the binding. Following mechanisms MUST be enforced:
    1. If the SAVI-SEND solution decides to bind an address despite that the binding collides with an existing FCFS/DHCP address, a SEND NA MUST be sent by the SAVI device.
    2. If a SAVI device receives a SEND NA targeting at a local bound address by FCFS and DHCP, it MUST remove the binding, and announce the conflict to the host with the binding.
    3. If a static address bound manually collides with any exiting binding, the existing binding MUST be removed manually by administrator, and the conflict MUST be announced to the host with existing binding.

5.3.1.3. Conflict Announcement

If a host is prohibited from using a bound address, the violation MUST be announced to it, through delivering one (or more) Neighbor Advertisement message to the host.

5.3.2. Same Address on the Same Binding Anchor

A binding may be set up on the same binding anchor by multiple solutions. Generally, the binding lifetimes of different solutions are different. Potentially, if one solution requires to remove the binding, the node using the address may be taken the use right.

For example, a node performs DAD procedure after being assigned an address from DHCP, then the address will also be bound by SAVI-FCFS. If the SAVI-FCFS lifetime is shorter than DHCP lifetime, when the SAVI-FCFS lifetime expires, it will request to remove the binding. If the binding is removed, the node will not be able to use the address even the DHCP lease time doesn't expire.

The solution proposed is to keep a binding as long as possible. A binding is kept until it has been required to be removed by all the solutions that ever set up it.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2. Informative References

[I-D.ietf-savi-dhcp]
Bi, J., Wu, J., Yao, G., and F. Baker, "SAVI Solution for DHCP", draft-ietf-savi-dhcp-06 (work in progress), September 2010.

[I-D.ietf-savi-fcfs]
Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS-SAVI: First-Come First-Serve Source-Address Validation for Locally Assigned Addresses", draft-ietf-savi-fcfs-05 (work in progress), October 2010.

[I-D.ietf-savi-framework]
Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, "Source Address Validation Improvement Framework", draft-ietf-savi-framework-01 (work in progress), October 2010.

[I-D.ietf-savi-send]
Bagnulo, M. and A. Garcia-Martinez, "SEND-based Source-Address Validation Implementation",

draft-ietf-savi-send-04 (work in progress), October 2010.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

Appendix A. Contributors and Acknowledgments

Thanks to Christian Vogt, Eric Nordmark, Marcelo Bagnulo Braun and Jari Arkko for their valuable contributions.

Authors' Addresses

Jun Bi
CERNET
Network Research Center, Tsinghua University
Beijing 100084
China

Email: junbi@cernet.edu.cn

Guang Yao
Tsinghua University
Network Research Center, Tsinghua University
Beijing 100084
China

Email: yaog@netarchlab.tsinghua.edu.cn

Joel M. Halpern
Newbridge Networks Inc

Email: jmh@joelhalpern.com

Eric Levy-Abegnoli
Cisco Systems
Village d'Entreprises Green Side - 400, Avenue Roumanille
Biot-Sophia Antipolis - 06410
France

Email: elevyabe@cisco.com